Available online at www.sciencedirect.com

ScienceDirect

NEUROCOMPUTING

Neurocomputing 69 (2006) 2309–2319

www.elsevier.com/locate/neucom

Genetic tracker with neural network for single and multiple target tracking

Ilke Turkmen[a], Kerim Guney[b,*], Dervis Karaboga[c]

[a]*Erciyes University, Civil Aviation School, Department of Aircraft Electrical and Electronics, 38039 Kayseri, Turkey*
[b]*Erciyes University, Faculty of Engineering, Department of Electronics Engineering, 38039 Kayseri, Turkey*
[c]*Erciyes University, Faculty of Engineering, Department of Computer Engineering, 38039 Kayseri, Turkey*

Received 20 July 2004; received in revised form 20 April 2005; accepted 27 April 2005
Available online 26 January 2006
Communicated by S. Fiori

Abstract

In this paper, a genetic tracker (GT) with neural network (NN) (GT-NN) is presented for single and multiple target tracking. The data association problem formulated as an *N*-dimensional assignment problem is solved using genetic algorithm. The incorporation of a NN into the GT is then proposed to increase its tracking performance. Performance evaluations of the GT, the GT-NN, the probabilistic data association filter, and the joint probabilistic data association filter are presented using simulation studies. Nine different tracking scenarios are considered for this evaluation. It has been observed that the estimation results of the GT-NN are better than those of the GT, the probabilistic data association filter, and the joint probabilistic data association filter.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Target tracking; Data association; Genetic algorithm; Neural networks

1. Introduction

Target tracking is an important problem with wide applications in both military and civilian areas such as ground target tracking, air traffic control, space surveillance, radar tracking in the presence of electronic countermeasures, image-based tracking of tissue cells, intelligent information retrieval, and visual tracking. The purpose of a target tracking system is to identify targets and then to estimate the states of the targets. The target state comprises position, velocity, and acceleration. In a cluttered environment, all of the received measurements may not arise from the targets of interest. Some of them may be from clutter or false alarm. Therefore, there always exist ambiguities in the association between the previous known targets and the measurements. A number of approaches [3,10,9,33,25,35,21,34,7,26,27,29,2,5,19,6,22], for solving this data association problem, have been reported in the literature, which can be grouped into two types: Bayesian and non-Bayesian data association.

Bayesian approach evaluates the probabilities of associations and uses them throughout the estimation process. The probabilistic data association filter (PDAF) [3] is one of the Bayesian approaches, commonly used for the single target tracking. It assumes that all measurements are in particular target extension gate and originated either from a target or clutter. If another target is persistently in this target extension gate, the results are poor and may be wrong. To account for the problem of more than one target in the cluttered environment, an extension of PDAF called joint probabilistic data association filter (JPDAF) [3,10] was proposed. It works without any prior information about the targets and clutters. In the JPDAF algorithm, the association probabilities are determined from the joint likelihood functions corresponding to the joint hypotheses associating all the returns to different permutations of the targets and clutter points. The real-time computation requirements of the JPDAF are very high and the computational complexity increases exponentially as the number of targets increases. To reduce this computational

*Corresponding author. Tel.: +90 352 437 57 55; fax: +90 352 437 57 84.
E-mail address: kguney@erciyes.edu.tr (K. Guney).

0925-2312/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2005.04.014

complexity, Fitzgerald [9] has proposed a simplified version of the JPDAF, called the cheap JPDAF (CJPDAF) algorithm. The CJPDAF method is very fast and easy to implement; however, the tracking performance of the CJPDAF also decreases in either dense target or cluttered environment. Details of the existing Bayesian data association algorithms can be found in [3]. These Bayesian data association algorithms have different levels of complexity and require vastly different computational efforts. Several methods based on different structures and architectures of the artificial neural networks (ANNs) [33,25,35] and the fuzzy inference systems [21,34,7] have also been presented for the calculation of the association probabilities.

The non-Bayesian approach is characterized by a hard association, such as assignment of a sequence of measurements together or of a measurement to a track, based on certain decision criterion [3,26,27,29]. The set of measurements from one scan constitutes a list. The assignment of a set of measurements from N times into sequences amounts to having N lists and finding sequences with one element from each list. This is an N-dimensional assignment problem, which is known to be NP-hard once N is greater than two. It is well known that solving such a constrained optimization problem is intractable. Genetic algorithm (GA) [20,8] is one technique which has demonstrated significant potential as a robust and efficient method for solving NP-hard problems. A number of methods [2,5,19,6] based on the GA have been proposed for target tracking. Angus et al. [2] and Carrier et al. [5] have described a method based on the GA for data association within single scan. Hillis [19] has proposed a technique to recast the multiscan assignment problem as a scheduling problem, in which the GA searches for an ordering of reports that produces good assignments. In [6], a dynamic fuzzy mutation GA has been presented to solve multidimensional assignment problem. In the genetic tracker (GT), the target states are updated by using Kalman filter [22]. The Kalman filtering is effective for simple scenarios such as in a clutterless environment or a single sensor tracking a single target. However, under dense target environment, extraneous sensor reports may be incorrectly used by the Kalman filter for track update, thus resulting in degraded performance, possibly loss of track may occur. In this paper, the inaccuracies in the estimation of Kalman filter are corrected by using ANN.

Ability and adaptability to learn, generalizability, smaller information requirement, fast real-time operation, noise elimination, and ease of implementation features have made ANNs popular in the last decade. In this paper, we proposed a hybrid system that combines the GT estimation capability with the NN learning capability to achieve the target tracking task without requiring an excessive computing resource. The data association problem is first formulated as an N-dimensional assignment problem and then solved using the GA. Next, ANN is incorporated into the GT to reduce the estimation error of the GT. In previous works [28,30,31,32,24,12,13,14,11,15,16,23,1], we successfully introduced ANNs [28,30,31,32,24,12,13,14,11,15,16] and GAs [23,1] for calculating accurately the various parameters of the triangular, rectangular and circular microstrip antennas, and pyramidal horn antennas. In the following sections, the JPDAF and the GT with the NN (GT-NN) proposed in this paper are described briefly, and the simulation studies are then presented.

2. The joint probabilistic data association filter (JPDAF)

The JPDAF [3,10] is an approach to multiple target tracking (MTT) that eliminates much of the complexity and computational cost usually associated with sophisticated MTT methods. The essence of the method is the computation of association probabilities for every track with every measurement in the present update, and the subsequent use of those probabilities as the weighting coefficients in the formation of a weighted average measurement for updating each target. For track t, the estimated state vector of the Kalman filter can be written as

$$\hat{x}_t(k|k) = \hat{x}_t(k|k-1) + K_t(k)\upsilon_t(k), \tag{1}$$

where $\hat{x}_t(k|k-1)$ is the predicted state vector, $K_t(k)$ is the Kalman gain, and $\upsilon_t(k)$ is the combined innovation. The combined innovation is given by

$$\upsilon_t(k) = \sum_{j=1}^{m_t(k)} \beta_{tj}\upsilon_{tj}(k), \tag{2}$$

where $m_t(k)$ is the number of validated measurements for track t, β_{tj} is the probability of association of track t with measurement j, and $\upsilon_{tj}(k)$ is the vector of measurement residuals (innovations) of track t and return j. The measurement innovation term, $\upsilon_{tj}(k)$, is defined by

$$\upsilon_{tj}(k) = z_j(k) - H_t(k)\hat{x}_t(k|k-1), \tag{3}$$

where $z_j(k)$ is the set of the all validated measurements received at the time k and $H_t(k)$ is the measurement matrix for target t.

The error covariance associated with the estimation given by Eq. (1) can be expressed as

$$\hat{P}_t(k|k) = \beta_{t0}(k)\hat{P}_t(k|k-1) + [1-\beta_{t0}(k)]P_t^c(k|k) + \tilde{P}_t(k) \tag{4}$$

with

$$P_t^c(k|k) = [I - K_t(k)H_t(k)]\hat{P}_t(k|k-1) \tag{5}$$

and

$$\tilde{P}_t(k) = K_t(k)\left[\sum_{j=1}^{m_t(k)} \beta_{tj}(k)\upsilon_{tj}(k)\upsilon_{tj}^{\mathrm{T}}(k) - \upsilon_t(k)\upsilon_t^{\mathrm{T}}(k)\right]K_t^{\mathrm{T}}(k), \tag{6}$$

where β_{t0} is the probability that none of the measurements within the validation gate of track t are correct, I is the identity matrix, and the superscript T indicates matrix transposition.

The error covariance of the predicted state is written as

$$\hat{P}_t(k|k-1) = F_t(k-1)\hat{P}_t(k-1|k-1)F_t^{\mathrm{T}}(k-1) + Q_t(k-1), \tag{7}$$

where $F_t(k)$ is the state transition matrix and $Q_t(k)$ is the process noise covariance matrix.

The Kalman filter gain is

$$K_t(k) = \hat{P}_t(k|k-1)H_t^{\mathrm{T}}(k)S_t^{-1}(k), \tag{8}$$

where $S_t(k)$ is the residual covariance given by

$$S_t(k) = H_t(k)\hat{P}_t(k|k-1)H_t^{\mathrm{T}}(k) + R_t(k), \tag{9}$$

where $R_t(k)$ is the measurement error variance.

The association probabilities β_{tj} in the JPDAF are determined by considering every possible hypothesis as to the association of the new measurements with the existing tracks. The computational complexity for the joint probabilities increases exponentially as the number of targets increases.

The problem is to find a method for target tracking that is as simple as possible, but the tracks estimated by using the method must be in good agreement with the true tracks. In this paper, a method based on the ANN and GA for efficiently solving this problem is presented. First, the data association problem, formulated as an N-dimensional assignment problem, is solved using the GA, then we incorporate the multilayered perceptron (MLP) [18] into the GT to improve the estimation results of GT.

3. GT-NN

In this study, the GA is used to find the most likely set of N-tuples such that each measurement is assigned to one and only one target, or declared false and each target receives at most, one measurement from each scan. The data association problem can be formulated as the following N-dimensional assignment problem:

minimize:

$$\sum_{i_1=0}^{M_1}\cdots\sum_{i_N=0}^{M_N} c_{i_1,\ldots,i_N}\chi_{i_1,\ldots,i_N},$$

subject to:

$$\sum_{i_2=0}^{M_2}\cdots\sum_{i_N=0}^{M_N} \chi_{i_1,\ldots,i_N} = 1, \quad i_1 = 1,\ldots,M_1,$$

$$\sum_{i_1=0}^{M_1}\cdots\sum_{i_{k-1}=0}^{M_{k-1}}\sum_{i_{k+1}=0}^{M_{k+1}}\cdots\sum_{i_N=0}^{M_N} \chi_{i_1,\ldots,i_N} = 1, \tag{10}$$

for $i_k = 1,\ldots,M_k$ and $k = 2,\ldots,N-1$,

$$\sum_{i_1=0}^{M_1}\cdots\sum_{i_{N-1}=0}^{M_{N-1}} \chi_{i_1,\ldots,i_N} = 1, \quad i_N = 1,\ldots,M_N,$$

$\chi_{i_1,\ldots,i_N} \subset \{0,1\}$, for all $i_1,\ldots,i_N$,

where $c_{i_1,\ldots,i_N}$ is the cost of associating the N-tuples of measurements $(i_1,\ldots,i_N)$ to track t, M is the number of the measurements in each scan, and $\chi_{i_1,\ldots,i_N}$ is the binary variable indicating the association of the N-tuples, and takes the values 0 or 1. The cost $c_{i_1,\ldots,i_N}$ can be written as the cumulative negative log-likelihood ratio given by

$$c_{i_1,\ldots,i_N} = -\sum_{m=1}^{N} \ln L_{i_m}(i_1,\ldots,i_m) \tag{11}$$

with the negative log-likelihood ratio of the track-measurement sequence $i_1,\ldots,i_m$ is

$$-\ln L_{i_m}(i_1,\ldots,i_m) = \frac{1}{2}[z_{i_m} - \hat{z}_{i_1,\ldots,i_m}]^{\mathrm{T}}S^{-1}_{i_1,\ldots,i_m}[z_{i_m} - \hat{z}_{i_1,\ldots,i_m}] + \ln\frac{\lambda_{\mathrm{e}}|2\pi S_{i_1,\ldots,i_m}|^{1/2}}{P_{\mathrm{D}}}, \tag{12}$$

where z is the measurement associated with track t at scan m, $\hat{z}$ is the predicted measurement from target t with covariance S, λ_{e} is the spatial density of the false alarms, and P_{D} is the detection probability.

The cost function for the GA is defined as

$$E = \text{minimize}\left(\sum_{i_1=0}^{M_1}\cdots\sum_{i_N=0}^{M_N} c_{i_1,\ldots,i_N}\chi_{i_1,\ldots,i_N}\right). \tag{13}$$

The value of the cost is mapped into a fitness value so as to fit into the GA. The better the solution encoded by a string, the higher the fitness. Minimizing E means getting a maximum fitness value in the genetic search process.

It is clear from Ref. [3] that the estimation error is determined primarily by the state space model and the measurement model. Since the estimation and the prediction vectors are closely related to the tracker, and hence to the estimation error, therefore they are suitable to be the inputs of the NN. Thus, the inputs of the NN are the position difference (δ_1) between the measurement and the estimation vectors, the position difference (δ_2) between the estimation and the prediction vectors, and the velocity difference (δ_3) between the estimation and the prediction vectors. The output of the NN is the estimation correction. Fig. 1 shows a block diagram of the approach proposed in this paper. The proposed approach can be called as GT-NN.

4. Simulations

In this section, several simulation examples are presented to illustrate the implementation of the proposed tracker that incorporates the MLP network into the GT, and to compare its performance with those of the GT, the PDAF, and the JPDAF. Nine different test scenarios are considered for this purpose. The trajectories of single

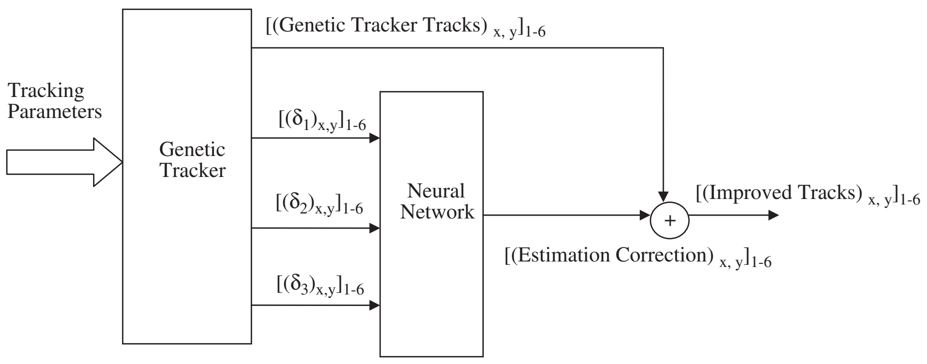

[(Genetic Tracker Tracks) x, y]1-6
Tracking Parameters
Genetic Tracker
[(δ1)x,y]1-6
[(δ2)x,y]1-6
[(δ3)x,y]1-6
Neural Network
+
[(Improved Tracks) x, y]1-6
[(Estimation Correction) x, y]1-6

Fig. 1. Block diagram of GT-NN.

manoeuvring targets in scenarios 1 and 2, two crossing targets in scenario 3, three crossing targets in scenario 4, four crossing targets in scenario 5, six crossing targets in scenario 6, two parallel targets in scenario 7, three parallel targets in scenario 8, and four parallel targets in scenario 9 are shown in Figs. 2–10, respectively. The trajectory of the manoeuvring target in scenario 1 represents a trajectory obtained from high-performance commercial aircraft [4]. The initial positions and velocities of the target in scenario 1 were chosen as [45.70 km, −42.86 km, −0.15 km/s, 0 km/s]. The trajectory of manoeuvring target in scenario 2 represents medium bomber flying [4] at high speed with good manoeuvrability. The initial positions and velocities of the target in scenario 2 were chosen as [60.94 km, 60.94 km, −0.16 km/s, −0.16 km/s]. The trajectories in scenarios 3–9 are similar to the ones widely used in the literature. The initial positions and velocities of crossing targets are listed in the first two rows, in the first three rows, in the first four rows, and in all rows of Table 1 for the scenarios 3, 4, 5, and 6, respectively. The initial positions and velocities of parallel targets are also given in the first two rows, in the first three rows, and in all rows of Table 2 for the scenarios 7, 8, and 9, respectively. The discretized state equation for each target is given by

$$X_t(k+1) = F_t(k)X_t(k) + G_t(k)w_t(k), \tag{14}$$

where X is the target state vector, F and G are known matrices describing the dynamics of the target, and w is a vector of zero-mean Gaussian noise uncorrelated with any such noise vector at a different instant of time. The state vector can be written as

$$X_t(k) = [x \;\; y \;\; \dot{x} \;\; \dot{y}]^{\mathrm{T}}, \tag{15}$$

where x and y are the position components, and $\dot{x}$ and $\dot{y}$ are the velocity components of the target t at time k. $F_t(k)$ and $G_t(k)$ are given by

$$F_t(k) = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad G_t(k) = \begin{bmatrix} T^2/2 & 0 \\ 0 & T^2/2 \\ T & 0 \\ 0 & T \end{bmatrix}, \tag{16}$$

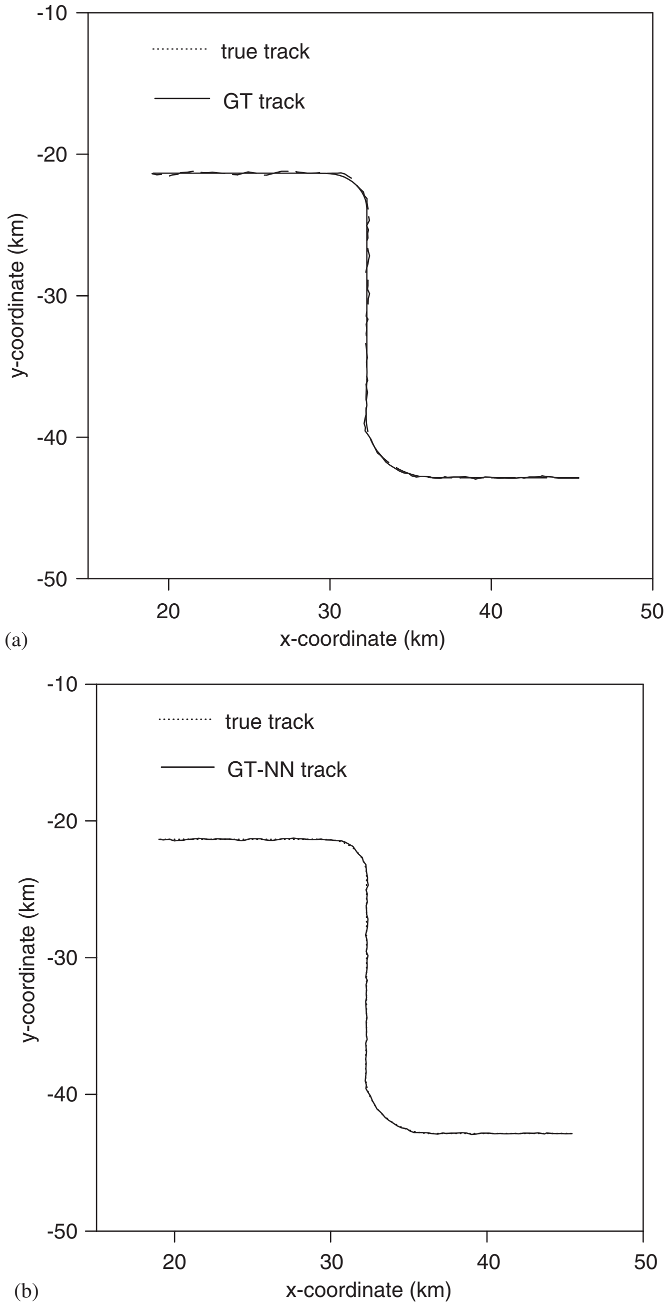
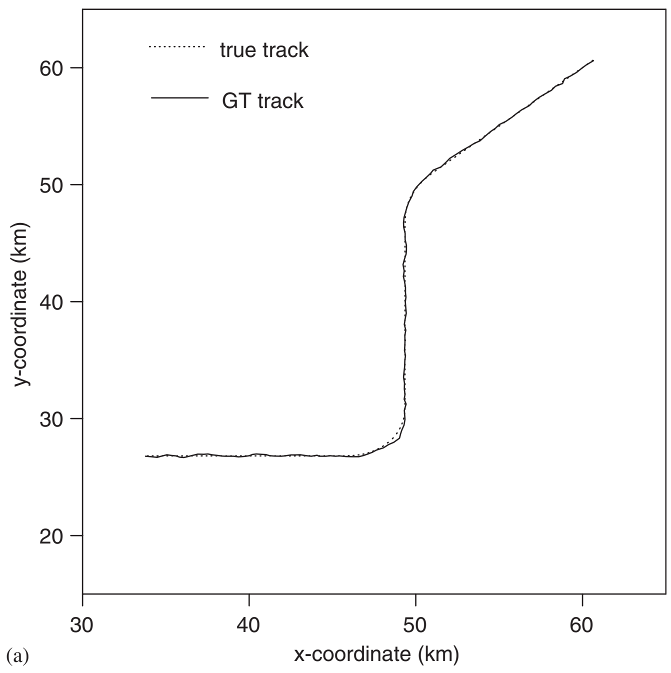


Fig. 2. Tracking single manoeuvring target in scenario 1 using (a) GT and (b) GT-NN.



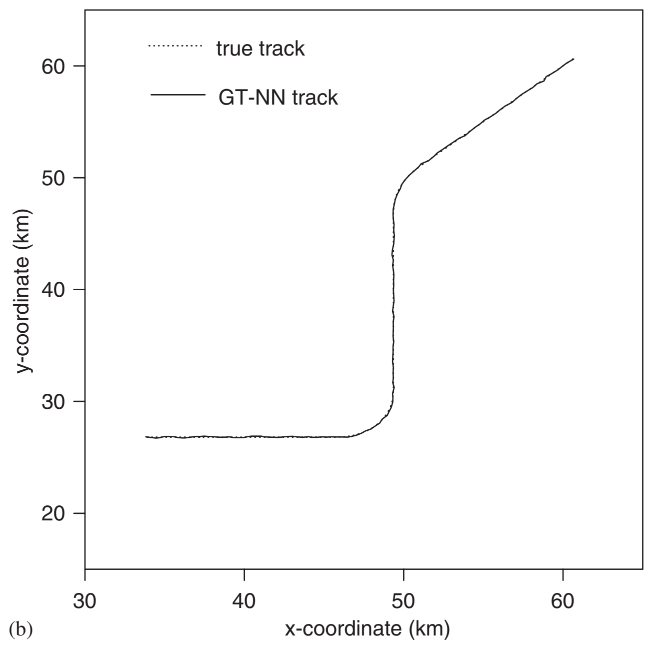
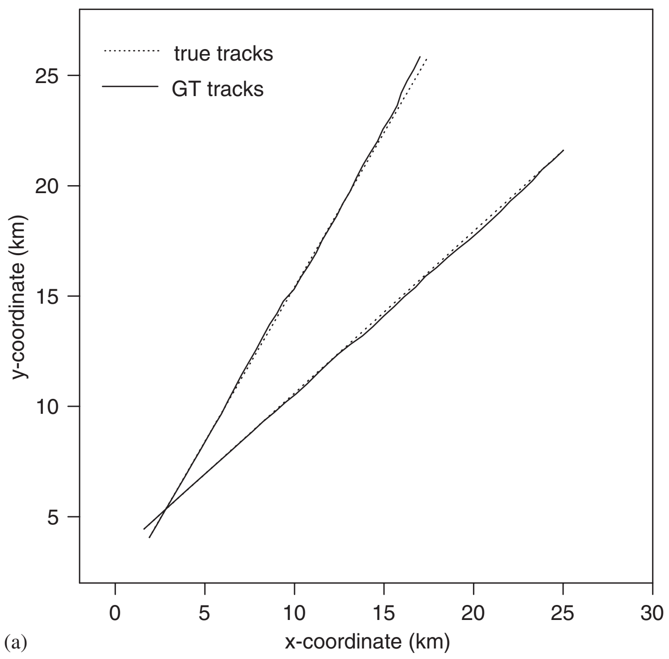


Fig. 3. Tracking single manoeuvring target in scenario 2 using (a) GT and (b) GT-NN.



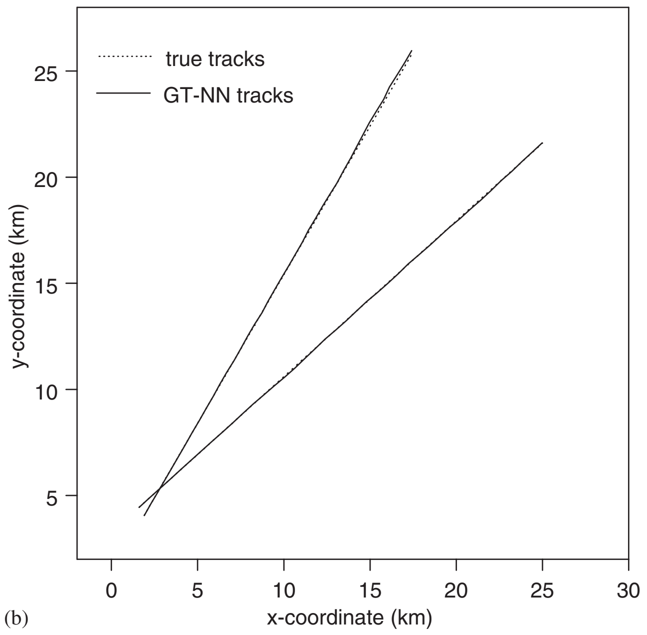


Fig. 4. Tracking two crossing targets in scenario 3 using (a) GT and (b) GT-NN.

where T is the sampling interval. In the simulation the sampling interval was assumed to be 1 s. The covariance matrix of the process noise is given by

$$Q_t(k) = \begin{bmatrix} (\sigma_x^t(k))^2 & 0 \\ 0 & (\sigma_y^t(k))^2 \end{bmatrix}. \quad (17)$$

For single manoeuvring targets in scenarios 1 and 2, the associated variances were chosen as

$(\sigma_x^t(k))^2 = 0.005\,\text{km}^2\,\text{s}^{-4}$,

$(\sigma_y^t(k))^2 = 0.005\,\text{km}^2\,\text{s}^{-4}$.

For multiple targets in scenarios 3–9, the associated variances were chosen as

$(\sigma_x^t(k))^2 = 1 \times 10^{-6}\,\text{km}^2\,\text{s}^{-4}$,

$(\sigma_y^t(k))^2 = 1 \times 10^{-6}\,\text{km}^2\,\text{s}^{-4}$.

The measurement model is given by

$$z(k) = H(k)X_t(k) + v(k), \quad (18)$$

where $z(k)$ is the measurement vector and v is a zero-mean Gaussian noise vector independent of w_t. It was assumed

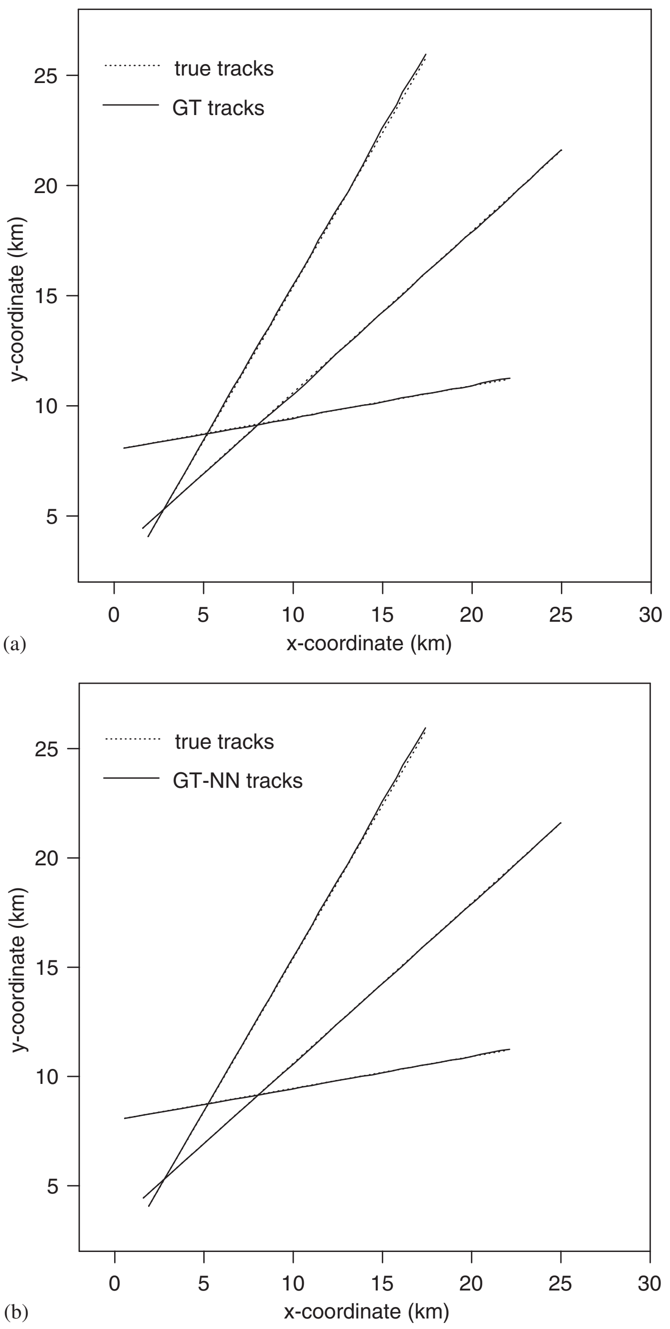


Fig. 5. Tracking three crossing targets in scenario 4 using (a) GT and (b) GT-NN.

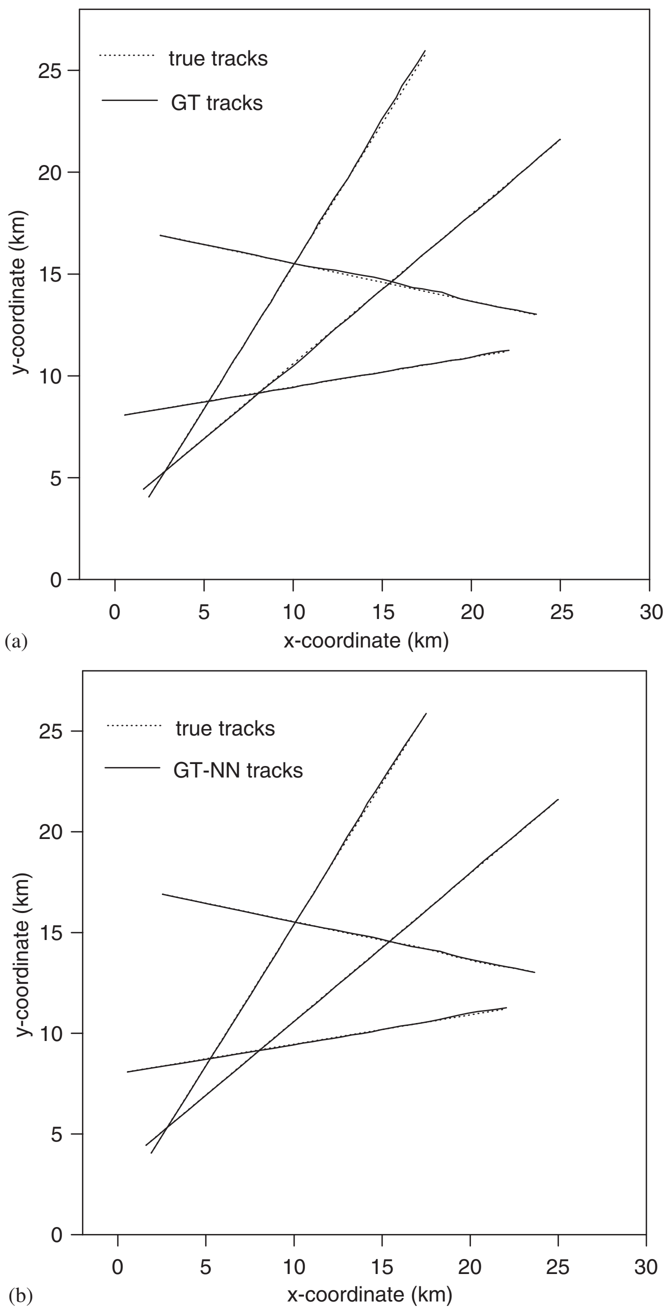


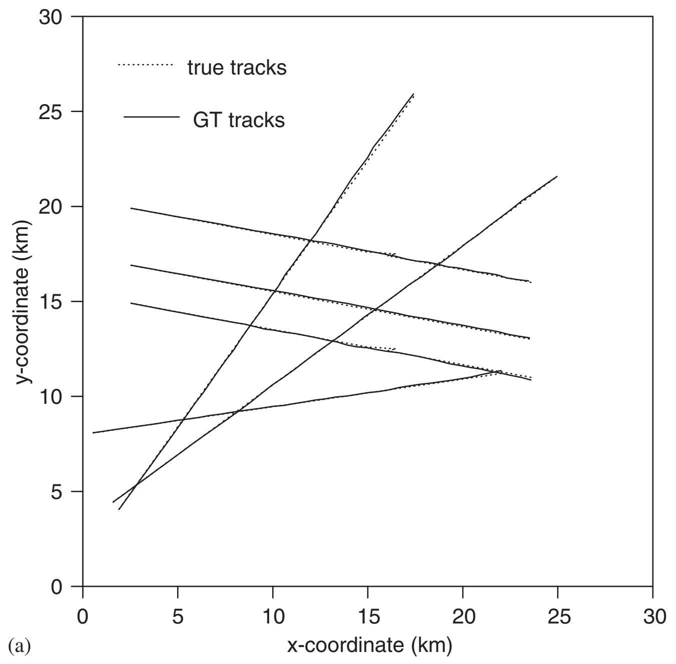
Fig. 6. Tracking four crossing targets in scenario 5 using (a) GT and (b) GT-NN.

that only position measurements are available so that

$$H(k) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{for all } k.$$

The covariance matrix of measurement noise $v(k)$ was $R(k) = \text{diag}(0.1, 0.1)\,\text{km}^2$ assuming all the measurement noise to be uncorrelated. The probability of detection was selected as 0.9.

In the simulation studies, there is one target in scenarios 1 and 2, and there are two targets in scenarios 3 and 7, three targets in the scenarios 4 and 8, four targets in scenarios 5 and 9, and six targets in scenario 6. The sensors can receive five measurements at every scan for single targets in scenarios 1 and 2, and eight measurements for multiple targets in scenarios 3–9. Therefore, there are four, five, six, and two clutter points at every scan for scenarios (1, 2, 5, and 9), (4 and 8), (3 and 7), and (6), respectively. The clutter points are uniformly located in the measurement space with an average value about four clutter points per validation gate for single target, and about two clutter points per validation gate for multiple targets. The false alarm rate of the sensor is chosen as to give the selected number of clutter points. The scan number in every sliding



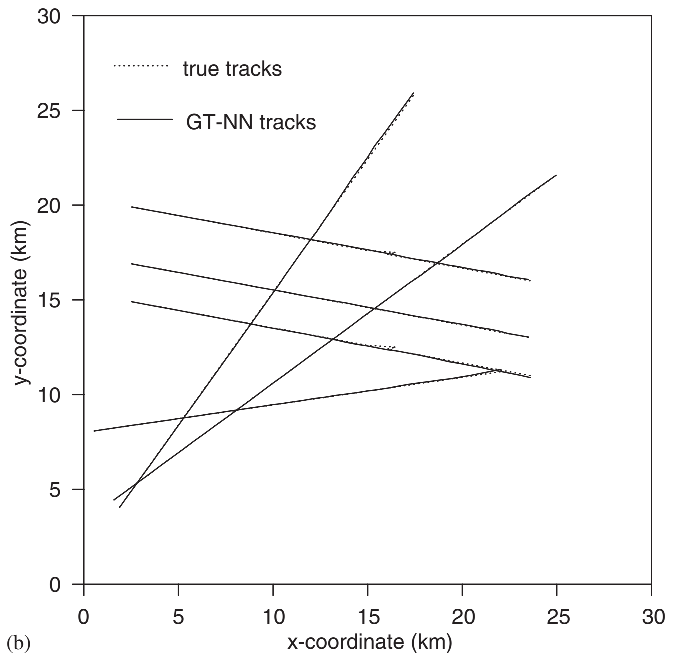
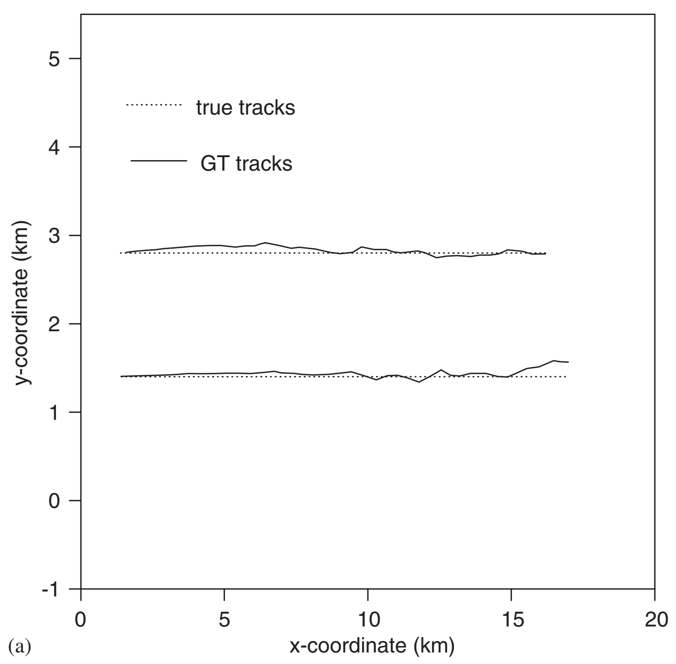


Fig. 7. Tracking six crossing targets in scenario 6 using (a) GT and (b) GT-NN.



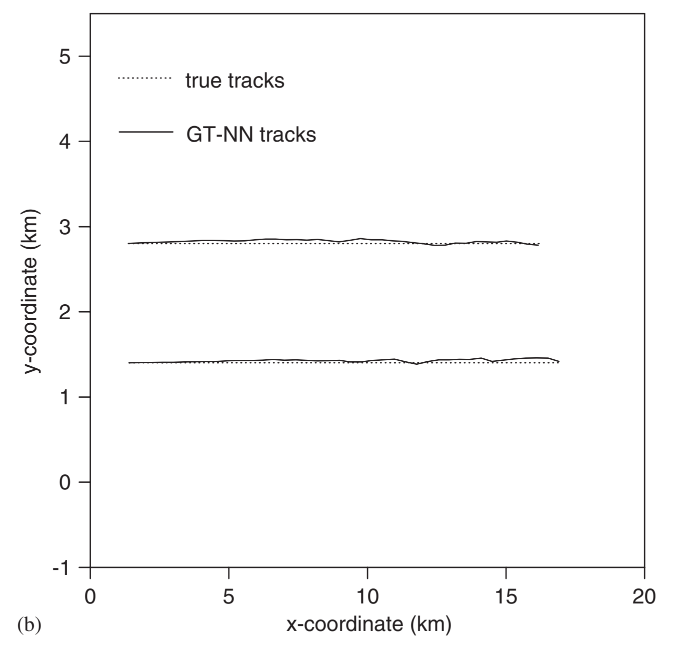
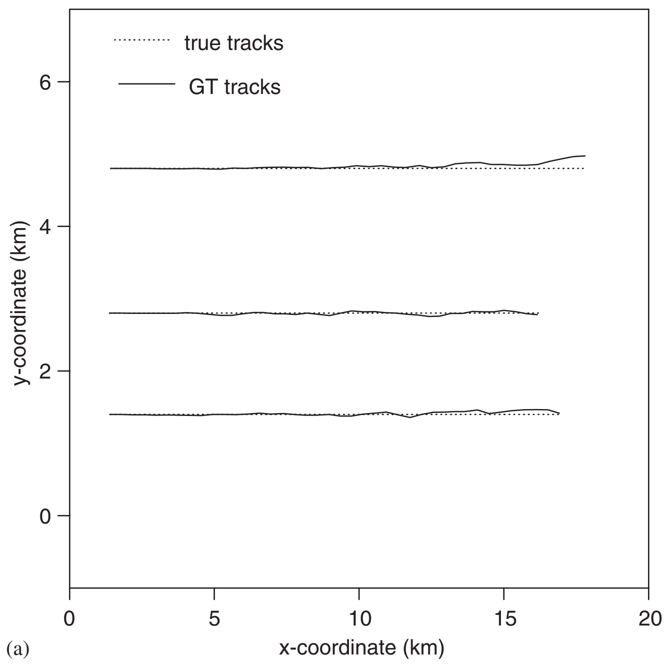


Fig. 8. Tracking two parallel targets in scenario 7 using (a) GT and (b) GT-NN.

window is 10 for single targets, and 5 for multiple targets. In the GA, we used the permutation coding. This representation can be explained for scenario 3 as follows: For target one in scenario 3, if the corresponding sequence consists of the first measurement of scan 1, the second measurement of scan 2, the fourth measurement of the scan 3, the fifth measurement of the scan 4, the eighth measurement of the scan 5; and for target two, the corresponding sequence consists of the sixth measurement of the scan 1, the fourth measurement of the scan 2, the seventh measurement of the scan 3, the fourth measurement of the scan 4, the fifth measurement of the scan 5, then the chromosome is a string [1, 2, 4, 5, 8, 6, 4, 7, 4, 5]. Similar representations were used for other scenarios. There are 10 elements in the coded strings for scenarios (1, 2, 3, and 7), 15 elements in the coded strings for scenarios (4 and 8), 20 elements in the coded strings for scenarios (5 and 9), and 30 elements in the coded strings for scenario (6). The fitness of a string is calculated by using the following function:

$$F = \begin{cases} 0, & \text{if the constraint is violated,} \\ -E, & \text{otherwise.} \end{cases} \tag{19}$$



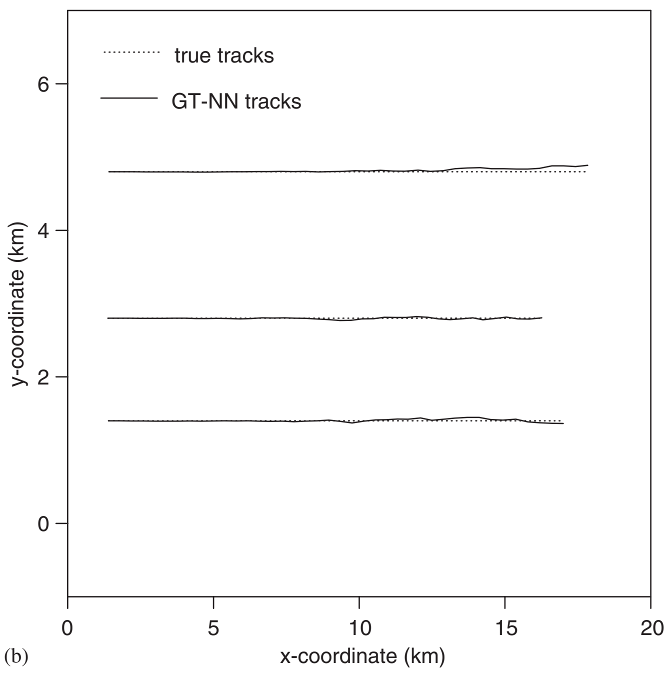
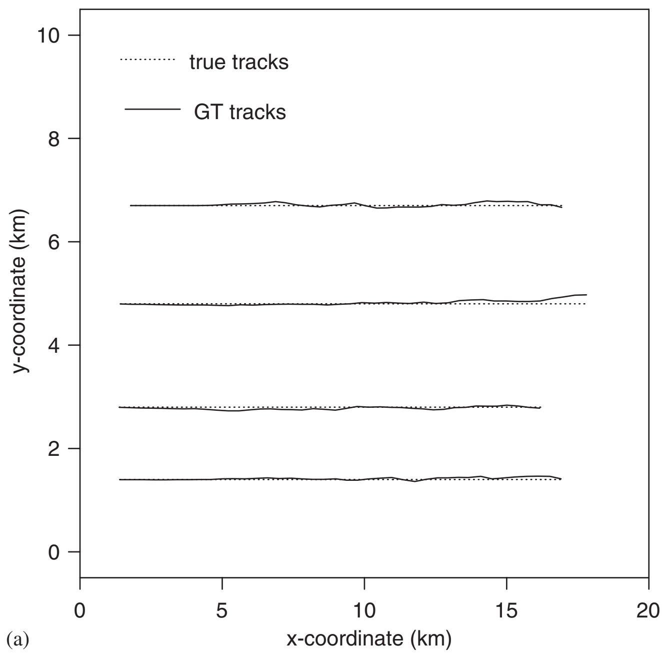


Fig. 9. Tracking three parallel targets in scenario 8 using (a) GT and (b) GT-NN.



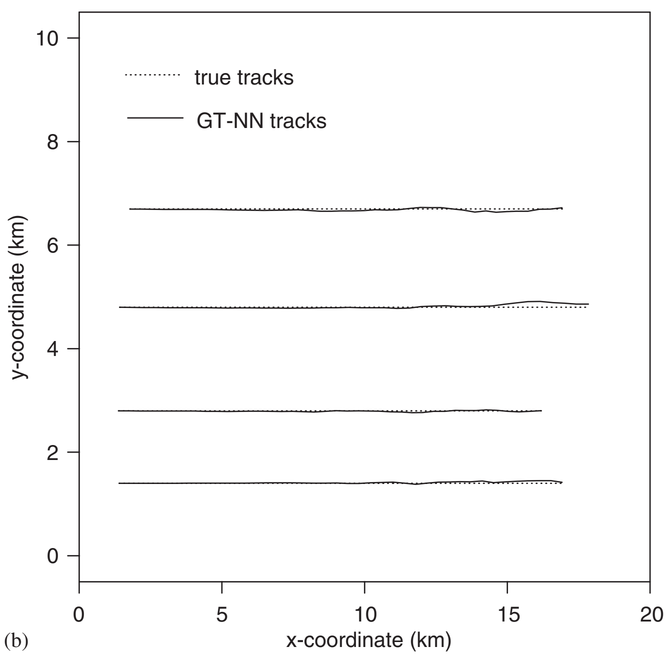


Fig. 10. Tracking four parallel targets in scenario 9 using (a) GT and (b) GT-NN.

The GA used in this paper employed the tournament selection scheme and multi-point crossover for the reproduction and the crossover operation, respectively. The population size was chosen as 100. The crossover rate and the mutation rate were taken as 0.8 and 0.15, respectively.

In the course of developing an ANN model, the architecture of the NN and the training algorithm are the two most important factors. ANNs have many structures and architectures [18]. The class of ANN and/or architecture selected for a particular model implementation

Table 1
Initial positions and velocities for crossing targets in scenarios 3–6

Crossing targets	x (km)	y (km)	$\dot{x}$ (km/s)	$\dot{y}$ (km/s)
1	1.5	3.5	0.40	0.56
2	1.0	4.0	0.60	0.44
3	0.0	8.0	0.55	0.08
4	2.0	17.0	0.54	−0.10
5	2.0	20.0	0.54	−0.10
6	2.0	15.0	0.54	−0.10

Table 2
Initial positions and velocities for parallel targets in scenarios 7–9

Parallel targets	x (km)	y (km)	$\dot{x}$ (km/s)	$\dot{y}$ (km/s)
1	1.01	1.41	0.41	0.001
2	1.02	2.82	0.39	0.002
3	1.01	4.81	0.43	0.001
4	1.39	6.72	0.40	0.003

depends on the problem to be solved. After several experiments using different architectures coupled with different training algorithms, in this paper, the MLP NN architecture was used to model accurately the estimation error of the GT. MLP has a simple layer structure in which successive layers of neurons are fully interconnected, with connection weights controlling the strength of the connections. Despite its limited complexity it is one of the most extensively used ANN architecture because of its well-known general approximation capabilities. In this paper, the MLPs were trained with the Levenberg–Marquardt algorithm [17]. The Levenberg–Marquardt algorithm is a least-squares estimation method based on the maximum neighborhood idea. It combines the best features of the Gauss–Newton technique and the steepest–descent method, but avoids many of their limitations. In particular, it generally does not suffer from the problem of slow convergence.

Typically, neural model development requires three sets of data, namely the training data, the validation data, and the test data. For target tracking applications, there are two types of data generators, namely measurement and simulation. The selection of a data generator depends on the application and the availability of the data generator. In this paper, 975 training and 300 validation data sets were obtained from the neighbourhood of the true trajectories. The true trajectories were corrupted with different clutter points, and training and validation data sets were collected by realizing the GT for the tracking scenarios. In the nine different test scenarios shown in Figs. 2–10, the true target trajectories were corrupted with random clutter points which were not used in training and validation. 165 and 150 test data sets were used for the 1 and 2 scenarios, respectively. 40 test data sets were used for each target in scenarios 3–9. Therefore, 80, 120, 160, 240, 80, 120, and 160 test data sets were used for scenarios 3–9, respectively.

Currently, there is no deterministic approach that can optimally determine the number of hidden layers and the number of neurons. A common practice is to take a trial and error approach which adjusts the hidden layers to strike a balance between memorization and generalization. After several trials, it was found in this paper that the most suitable network configuration was three hidden layers with eight neurons for the first and second hidden layers and four neurons for the third hidden layer. The training is stopped when the validation error begins to increase. The validation error started to increase at around epoch number of 1200. The input and output layers of the MLPs have linear transfer functions and the hidden layers have hyperbolic tangent functions. The input and output data sets were scaled between 0.0 and 1.0 before training. The seed number was fixed to 87. The value of μ of Levenberg–Marquardt algorithm was chosen as 0.2.

The tracking performances of the GT and the GT-NN are compared in Figs. 2–10 for 9 test scenarios. As it is seen from Figs. 2–10, the GT-NN tracks are closer to the true tracks than the tracks predicted by the GT for all scenarios. The results of the PDAF and the JPDAF are not shown in Figs. 2–10 for clarity but RMS tracking errors of the PDAF and the JPDAF are given in Table 3. Table 3 also gives the comparative performances of the PDAF, the JPDAF, the GT, and the GT-NN methods in terms of RMS tracking error. The percentage improvement obtained by using GT-NN is calculated as the ratio of the difference between the RMS errors of the GT-NN method and the competing method (PDAF, JPDAF, or GT) to the RMS error of the competing method. It is apparent from Table 3 that in all cases the results of the GT-NN are better than those of the PDAF, the JPDAF, and the GT methods. The RMS tracking error values clearly show that a significant improvement is obtained on the results of the PDAF, the JPDAF, and the GT methods. The average percentage improvements with respect to the PDAF, the JPDAF, and the GT are 45%, 46%, and 39%, respectively. The incorporation of a NN into the GT leads to good accuracy in tracking single and multiple targets.

The NN is a very powerful approach for building complex and non-linear relationship between a set of input and output data. Accurate, fast, and reliable GT-NN models can be developed from measured/simulated data. Once developed, these GT-NN models can be used in place of computationally intensive target tracking models to speed up target tracking. A prominent advantage of NN calculation is that, after proper training, NN completely bypasses the repeated use of complex iterative processes for new cases presented to it. The trained ANN model can be used during target tracking to provide instant answers to the task it learned.

5. Conclusion

The GT-NN technique, which combines the learning capability of the NN and the tracking capability of the GT, is proposed for single and multiple target tracking. In this technique, first, the data association problem formulated as an N-dimensional assignment problem is solved by using GA and then the inaccuracies in the estimation are corrected by the NN. The results of GT-NN technique show a better agreement with the original tracks than the results of the GT, the PDAF, and the JPDAF. The very good agreement between the GT-NN tracks and the

Table 3
Performance comparison of the PDAF, the JPDAF, the GT, and the GT-NN methods

Scenarios		Targets	RMS tracking errors (km)			Percentage improvement with respect to the PDAF and JPDAF (%)	Percentage improvement with respect to the GT (%)
			PDAF and JPDAF[a]	GT	Present method (GT-NN)		
1	One manoeuvring target	1	0.0973	0.0876	0.0533	45	39
2	One manoeuvring target	1	0.0952	0.0873	0.0524	45	40
3	Two crossing targets	1	0.0897	0.0871	0.0452	50	48
		2	0.0621	0.0595	0.0268	57	55
4	Three crossing targets	1	0.0740	0.0622	0.0461	38	26
		2	0.0320	0.0328	0.0212	34	35
		3	0.1071	0.0914	0.0685	36	25
5	Four crossing targets	1	0.0644	0.0509	0.0297	54	42
		2	0.0351	0.0338	0.0236	33	30
		3	0.1058	0.0912	0.0606	43	34
		4	0.0896	0.0711	0.0327	64	54
6	Six crossing targets	1	0.0598	0.0551	0.0227	62	59
		2	0.0311	0.0249	0.0199	36	20
		3	0.1010	0.0968	0.0420	58	57
		4	0.0568	0.0515	0.0258	55	50
		5	0.0712	0.0673	0.0397	44	41
		6	0.0785	0.0682	0.0367	53	46
7	Two parallel targets	1	0.0899	0.0772	0.0495	45	36
		2	0.0722	0.0653	0.0337	53	48
8	Three parallel targets	1	0.0689	0.0659	0.0493	28	25
		2	0.0290	0.0269	0.0210	28	22
		3	0.1028	0.0901	0.0449	56	50
9	Four parallel targets	1	0.0646	0.0526	0.0335	48	36
		2	0.0389	0.0379	0.0223	43	41
		3	0.0945	0.0901	0.0461	51	49
		4	0.0501	0.0433	0.0380	24	12

[a]PDAF is used for single target tracking (scenarios 1 and 2) and JPDAF is used for MTT (scenarios 3–9).

original tracks substantiates the validity of GT-NN technique.

References

[1] A. Akdagli, K. Guney, Effective patch radius expression obtained using a genetic algorithm for the resonant frequency of electrically thin and thick circular microstrip antennas, IEE Proceed. Microwaves Antennas Propagat. Pt.H 147 (2) (2000) 156–159.

[2] J. Angus, H. Zhou, C. Bea, L. Becket-Lemus, J. Klose, S. Tubbs, Genetic algorithms in passive tracking, Claremont Graduate School, Math Clinic Report, May 1993.

[3] Y. Bar-Shalom, X.R. Li, Multitarget-Multisensor Tracking: Principles and Techniques, YBS Publishing, 1995.

[4] W.D. Blair, G.A. Watson, T. Kirubarajan, Y. Bar-Shalom, Benchmark for radar allocation and tracking in ECM, IEEE Trans. Aerospace Electron. Systems 34 (4) (1998) 1097–1114.

[5] J. Carrier, J. Litva, H. Leung, T. To, Genetic algorithm for multiple target tracking data association, Proc. SPIE Acquisit. Track. Point. 2739 (1996) 180–190.

[6] G. Chen, L. Hong, A genetic based multi dimensional data association algorithm for multi sensor multi target tracking, Math. Comput. Model. 26 (4) (1997) 57–69.

[7] Y.M. Chen, H.C. Huang, Fuzzy logic approach to multisensor data association, Math. Comput. Simulat. 52 (2000) 399–412.

[8] L. Davis, Handbook of Genetic Algorithms, Von Nostrand Reinhold, New York, 1991.

[9] R.J. Fitzgerald, Development of practical PDA logic for multitarget tracking by microprocessor, in: Proceedings of the American Control Conference, Seattle, Washington, 1986, pp. 889–897.

[10] T.E. Fortmann, Y. Bar-Shalom, M. Scheffe, Sonar tracking of multiple targets using joint probabilistic data association, IEEE J. Ocean. Eng. EO-8 (1983) 173–184.

[11] S. Gultekin, K. Guney, S. Sagiroglu, Neural networks for the calculation of bandwidth of rectangular microstrip antennas, Appl. Computat. Electromagn. Soc. (ACES) J. (Special Issue: Neural Network Appl. Electromagn.) 18 (2) (2003) 46–56.

[12] K. Guney, M. Erler, S. Sagiroglu, Artificial neural networks for the resonant resistance calculation of electrically thin and thick rectangular microstrip antennas, Electromagnetics 20 (2000) 387–400.

[13] K. Guney, S. Sagiroglu, M. Erler, Comparison of neural networks for resonant frequency computation of electrically thin and thick rectangular microstrip antennas, J. Electromagn. Waves Appl. (JEWA) 15 (2001) 1121–1145.

[14] K. Guney, S. Sagiroglu, M. Erler, Generalized neural method to determine resonant frequencies of various microstrip antennas, Int. J.

RF Microwave Comput.-Aided Eng. (Special Issue: Appl. Artif. Neural Networks RF Microwave Design) 12 (2002) 131–139.

[15] K. Guney, N. Sarikaya, Artificial neural networks for calculating the input resistance of circular microstrip antennas, Microwave Opt. Technol. Lett. 37 (2003) 107–111.

[16] K. Guney, N. Sarikaya, Artificial neural networks for the narrow aperture dimension calculation of optimum gain pyramidal horns, Electr. Eng. 86 (2004) 157–163.

[17] M.T. Hagan, M. Menjah, Training feedforward networks with the Marquardt algorithm, IEEE Trans. Neural Networks 5 (6) (1994) 989–993.

[18] S. Haykin, Neural Networks: A Comprehensive Foundation, Macmillan College Publishing Company, New York, 1994.

[19] D.B. Hillis, Using a genetic algorithm for multi-hypothesis tracking, in: Proceedings of the Ninth International Conference on Tools with Artificial Intelligence, 1997, pp. 112–117.

[20] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, MI, 1975.

[21] O. Hossam, M. Farooq, T. Quach, Fuzzy logic approach to data association, Proc. SPIE 2755 (1996) 313–321.

[22] R.E. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME—J. Basic Eng. (1960) 35–45.

[23] D. Karaboga, K. Guney, N. Karaboga, A. Kaplan, Simple and accurate effective side length expression obtained by using a modified genetic algorithm for the resonant frequency of an equilateral triangular microstrip antenna, Int. J. Electron. (IJE) 83 (1) (1997) 99–108.

[24] D. Karaboga, K. Guney, S. Sagiroglu, M. Erler, Neural computation of resonant frequency of electrically thin and thick rectangular microstrip antennas, IEE Proc.-Microwaves, Antennas Propagat. Pt.H. 146 (1999) 155–159.

[25] H. Leung, Neural-network data association with application to multiple-target tracking, Opt. Eng. 35 (3) (1996) 693–700.

[26] K.R. Pattipati, S. Deb, Y. Bar-Shalom, R.B. Washburn, A new relaxation algorithm and passive sensor data association, IEEE Trans. Automatic Control 37 (2) (1992) 197–213.

[27] K.R. Pattipati, R.L. Popp, T. Kirubarajan, Survey of assignment techniques for multitarget tracking, in: Y. Bar-Shalom, W.D. Blair (Eds.), Multitarget-Multisensor Tracking: Applications and Advances, 2000.

[28] D.T. Pham, D. Karaboga, Intelligent Optimisation Techniques, Springer, London, UK, 2000.

[29] A. Poore, N. Rijavec, Multitarget tracking, multidimensional assignment problems, and Lagrangian relaxation, Proceedings of the SDI Panels on Tracking, August 1991, pp. 51–74.

[30] S. Sagiroglu, K. Guney, Calculation of resonant frequency for an equilateral triangular microstrip antenna with the use of artificial neural networks, Microwave Opt. Technol. Lett. 14 (1997) 89–93.

[31] S. Sagiroglu, K. Guney, M. Erler, Resonant frequency calculation for circular microstrip antennas using artificial neural networks, Int. J. RF Microwave Comput.-Aided Eng. 8 (1998) 270–277.

[32] S. Sagiroglu, K. Guney, M. Erler, Calculation of bandwidth for electrically thin and thick rectangular microstrip antennas with the use of multilayered perceptrons, Int. J. RF Microwave Comput.-Aided Eng. (Special Issue: Appl. Artif. Neural Networks RF Microwave Design) 9 (1999) 277–286.

[33] D. Sengupta, R.A. Iltis, Neural solution to the multitarget tracking data association problem, IEEE Trans. Aerospace Electron. Systems 4 (1) (1989) 96–108.

[34] R.P. Singh, W.H. Balley, Fuzzy logic applications to multi-sensor multitarget correlation, IEEE Trans. Aerospace Electron. Systems 33 (3) (1997) 752–769.

[35] F. Wang, J. Litva, T. Lo, E. Bosse, Performance of neural data associator, IEE Proc. Radar Sonar Navigat. 143 (2) (1996) 71–78.

Ilke Turkmen was born in Kayseri, Turkey, on December 26, 1976. She received B.S., M.S., and Ph.D. degrees in electronics engineering from Erciyes University, Kayseri, Turkey, in 1997, 1999, and 2005, respectively. Currently, she is a lecturer at the Department of Aircraft Electrical and Electronics of Civil Aviation School, Erciyes University. Her current research activities include neural networks, genetic algorithms, fuzzy inference systems, and their applications to target tracking.

Kerim Guney was born in Isparta, Turkey, on February 28, 1962. He received the B.S. degree from Erciyes University, Kayseri, in 1983, the M.S. degree from Istanbul Technical University, in 1988, and the Ph.D. degree from Erciyes University, in 1991, all in electronic engineering. From 1991 to 1995 he was an assistant professor and now is a professor at the Department of Electronic Engineering, Erciyes University, where he is working in the areas of optimization techniques (the genetic, the tabu search, the differential evolution, and the ant colony optimization algorithms), fuzzy inference systems, neural networks, their applications to antennas, microstrip and horn antennas, antenna pattern synthesis, and target tracking. He has published more than 175 journal and conference papers.

Dervis Karaboga was born on April 24, 1962, in K. Maraş, Turkey. He received the B.S. degree from Erciyes University, Kayseri, Turkey, in 1983, and the M.S. degree from Istanbul Technical University, in 1988, both in electronic engineering. He received the Ph.D. degree in systems engineering from the University of Wales, College of Cardiff, UK, in 1994. He is now a professor at the Department of Computer Engineering, Erciyes University. His research interests include modern heuristic optimization techniques (genetic, tabu search, simulated annealing, ant colony, and artificial immune algorithms), fuzzy systems, neural networks, digital filter design, antenna design, intelligent control, and robotics.